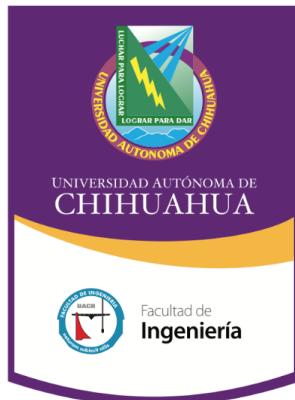


**UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA**  
**FACULTAD DE INGENIERÍA**



**CIRCUITOS LÓGICOS II**  
**GRUPO: 7HW1**

**PRÁCTICA NO. 6**  
**“SUMADOR DE 4-BITS”**

**EQUIPO**

*“sudo rm -rf / --no-preserve-root”*

**INTEGRANTES DEL EQUIPO**

JESÚS ALEJANDRO JIMÉNEZ HERNÁNDEZ  
OSCAR ALEJANDRO SERRANO PIZARRO  
EMILIO ALBERTO SALAS RIOS

**FECHA DE ENTREGA**

MARTES 11 DE OCTUBRE DE 2022

**DOCENTE**

JESÚS MANUEL MUÑOZ LARGUERO

## ÍNDICE

<b>INTRODUCCIÓN</b>	<b>3</b>
Objetivos	3
<b>DESARROLLO</b>	<b>4</b>
1. Configurar el Software Quartus II para que pueda funcionar con el CPLD MAX II.	4
2. Armar un circuito sumador de 4 bits completo.	4
3. Compilar y grabar en el CPLD el programa o el diagrama esquemático lógico, con el que funcionará nuestro decodificador.	5
<b>CONCLUSIÓN</b>	<b>8</b>
<b>BIBLIOGRAFÍA</b>	<b>9</b>

## INTRODUCCIÓN

En esta práctica se nos pide configurar el CPLD como un sumador (7483) de cuatro bits que cuente con la capacidad de mostrar el resultado ya sea a través de un display de siete segmentos, diodos LED o cualquier otra forma de mostrar el resultado. Se nos dio la oportunidad de realizarlo con el diagrama lógico o código VHDL.

### Objetivos

- Describir en lenguaje VHDL, Verilog o un con un diagrama lógico el circuito que haga la función de un sumador completo de 4 Bits.
- La salida de la suma deberá ser representada con LEDs o bien por medio de 4 displays de 7 segmentos.
- Armar el circuito y probar el funcionamiento del sumador.

## DESARROLLO

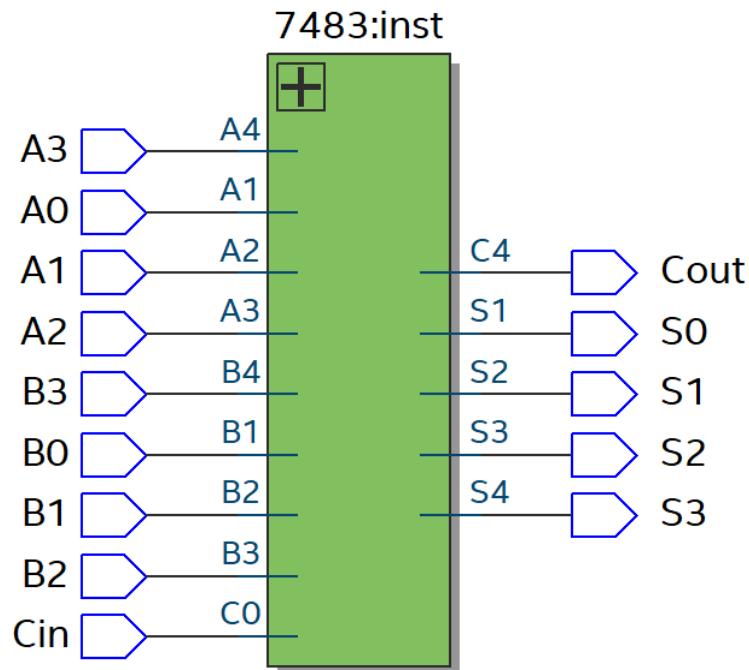
### 1. Configurar el Software Quartus II para que pueda funcionar con el CPLD MAX II.

La configuración del software está indicada dentro del documento que provee las instrucciones para la práctica, es por ello que tomamos la decisión de no incluirlo en el reporte de esta práctica. El documento puede ser encontrado en la bibliografía.

*NOTA: Es importante aclarar que durante esta práctica, el CPLD presentó fallas que aún no logramos solucionar, por lo que no fue posible mostrar y grabar el circuito en funcionamiento, sin embargo se incluye y muestran imágenes del circuito físico así como la implementación del mismo a través de Arduino para mostrar lo que debería hacer el circuito.*

### 2. Armar un circuito sumador de 4 bits completo.

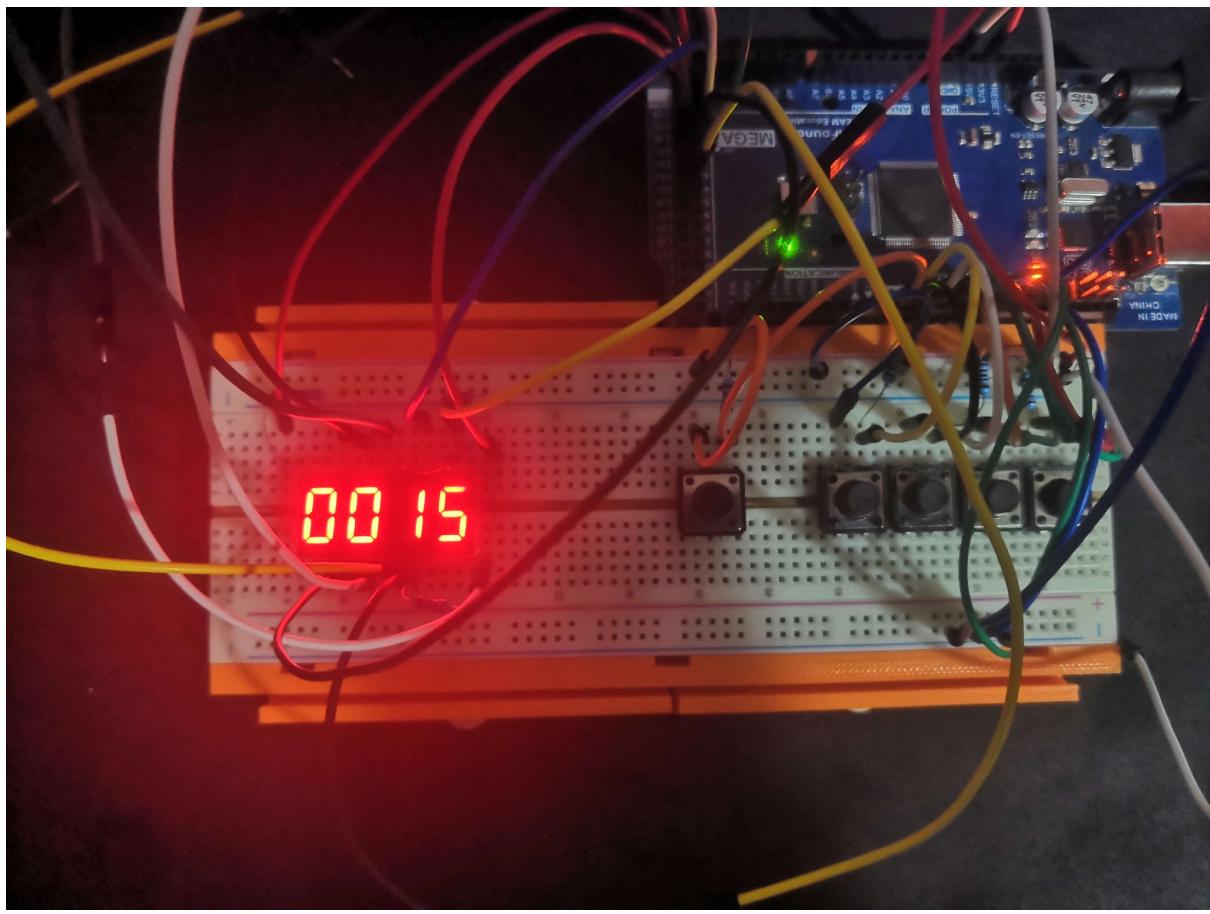
Este es el diagrama RTL del circuito físico, en el siguiente punto se muestra la configuración dentro del programa, esta imagen es únicamente didáctica.



Y finalmente, se muestra el circuito físico: <https://youtu.be/ByKUy7-I8Mg>

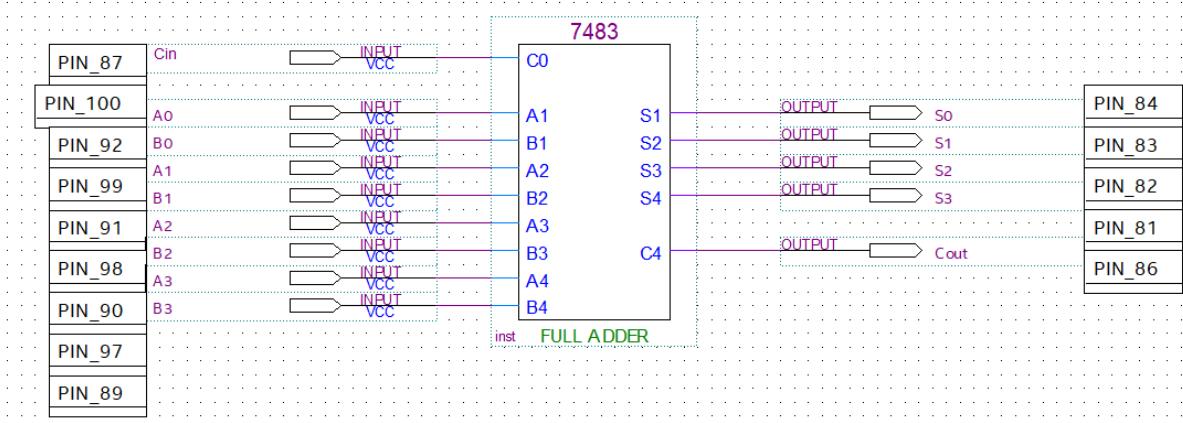
Se incluye el link al código de Arduino:

<https://github.com/gonnaflick/Circuitos-Logicos-II/tree/main/4-bit-adder>



### 3. Compilar y grabar en el CPLD el programa o el diagrama esquemático lógico, con el que funcionará nuestro decodificador.

En nuestro caso, se utilizó un display de siete segmentos de cátodo común, por lo que el decodificador que utilizamos para el circuito es el 7448.



Como podemos observar en esta imagen, se tiene el diagrama lógico del funcionamiento del programa que se instalará en el CPLD.

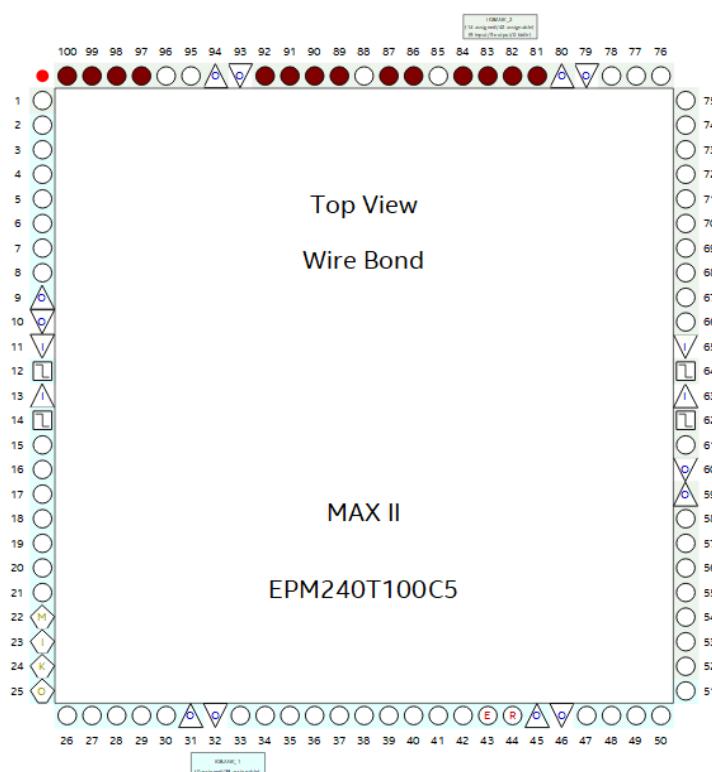
Continuando, la configuración de entrada es la siguiente:

- Pin 87 → Cin → Botón Bit 0 → 1 decimal para el carry
- Pin 100 → A0 → Botón Bit 0 → 1 decimal
- Pin 99 → A1 → Botón Bit 1 → 2 decimal
- Pin 98 → A2 → Botón Bit 2 → 4 decimal
- Pin 97 → A3 → Botón Bit 3 → 8 decimal
- Pin 92 → B0 → Botón Bit 0 → 1 decimal
- Pin 91 → B1 → Botón Bit 1 → 2 decimal
- Pin 90 → B2 → Botón Bit 2 → 4 decimal
- Pin 89 → B3 → Botón Bit 3 → 8 decimal

Las compuertas de salida se encuentran de la siguiente forma:

- Pin 84 → Salida A → Bit 0 → 1 decimal
- Pin 83 → Salida B → Bit 1 → 2 decimal
- Pin 82 → Salida C → Bit 2 → 4 decimal
- Pin 81 → Salida D → Bit 3 → 8 decimal
- Pin 86 → Cout → Bit 0 → 1 decimal

En esta imagen se muestra la configuración de pines necesaria para programar el CPLD.



Esta imagen nos muestra los pines de entrada anteriormente mencionados.

Node Name	Direction	Location	I/O Bank	I/O Standard	Reserved	Current Strength	Strict Preservation
in A0	Input	PIN_100	2	3.3-V L...efault)		16mA (default)	
in A1	Input	PIN_99	2	3.3-V L...efault)		16mA (default)	
in A2	Input	PIN_98	2	3.3-V L...efault)		16mA (default)	
in A3	Input	PIN_97	2	3.3-V L...efault)		16mA (default)	
in B0	Input	PIN_92	2	3.3-V L...efault)		16mA (default)	
in B1	Input	PIN_91	2	3.3-V L...efault)		16mA (default)	
in B2	Input	PIN_90	2	3.3-V L...efault)		16mA (default)	
in B3	Input	PIN_89	2	3.3-V L...efault)		16mA (default)	
in Cin	Input	PIN_87	2	3.3-V L...efault)		16mA (default)	
out Cout	Output	PIN_86	2	3.3-V L...efault)		16mA (default)	
out S0	Output	PIN_84	2	3.3-V L...efault)		16mA (default)	
out S1	Output	PIN_83	2	3.3-V L...efault)		16mA (default)	
out S2	Output	PIN_82	2	3.3-V L...efault)		16mA (default)	
out S3	Output	PIN_81	2	3.3-V L...efault)		16mA (default)	

## CONCLUSIÓN

En esta práctica se realizó un sumador con la capacidad de sumar cuatro bits de una entrada A y cuatro de una entrada B. A su vez se le permitía tomar en cuenta un carry anterior. Cabe aclarar que en la práctica debido al fallo con el CPLD se presentó la problemática de la adaptación del código en Arduino a VHDL, por lo que nos vimos forzados a simplificar el código en contraste con lo anteriormente realizado, sin embargo, presentamos como complemento el código de Arduino que muestra la lógica del circuito. Resumiendo, la tablilla actuará como el 7483, 7448 y 74HC573. Cuando la tablilla reciba un voltaje proveniente de alguno de los botones, a partir del sumador de cuatro bits (7483), determinará su valor y lo sumará, este luego retornara cuatro bits y un carry de salida. Estos juegos pasan por un decodificador (7448) que los pasara a valores para que este se conecte a un display de siete segmentos y finalmente el 74HC573 o bien el shift register, alternará los valores del display en dos displays para mostrar el resultado de la suma de forma decimal.

En contraste con el circuito armado en el diagrama lógico, debido a la falta de la última compuerta, no fue posible mostrar el circuito. A su vez, debido a que no se contaba con el último chip, no era posible utilizar el 7448 debido a que este mostraría los valores únicamente hasta el número 9, y luego en hexadecimal, cosa que no realizaba el circuito armado en arduino.

## BIBLIOGRAFÍA

Documentación de práctica:

[https://aulas3.uach.mx/pluginfile.php/1117707/mod\\_assign/introattachment/0/  
Practica%20%236%20-%20Sumador%204%20Bits.pdf?forcedownload=1](https://aulas3.uach.mx/pluginfile.php/1117707/mod_assign/introattachment/0/Practica%20%236%20-%20Sumador%204%20Bits.pdf?forcedownload=1)